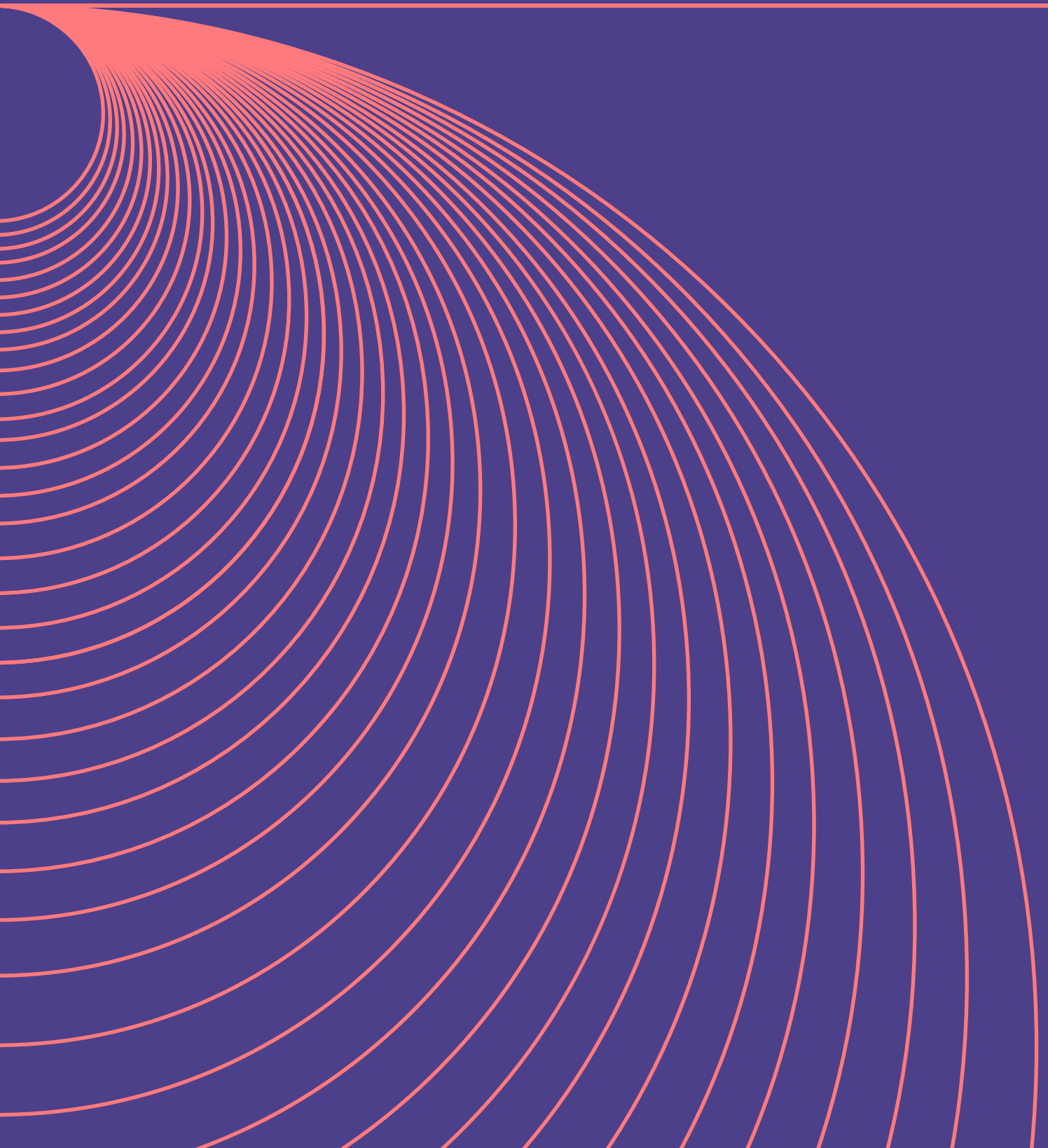


UXPin

Timeless UX Design Trends:

Responsive & Adaptive Web Design



UXPin

Timeless UX Design Trends:

Responsive & Adaptive Web Design

Copyright © 2015 by UXPin Inc.

All rights reserved. No part of this publication text may be uploaded
or posted online without the prior written permission of the publisher.

For permission requests, write to the publisher, addressed “Attention:
Permissions Request,” to hello@uxpin.com.

Index

M-Dot Sites Are Dead	7
Consistent Experience Across Devices	11
Why?	11
How?	12
Responsive and Adaptive Design	17
Responsive Design	17
RWD Best Practices	19
Adaptive Design	21
AWD Best Practices	22
Mobile-First Design	24
Mobile-First = Content-First	25
The Mobile-First Process	26
A Mobile-First Design Lesson	30
Set your content priorities	30
Design device-appropriate layouts yourself	34

Real Examples of Consistent UX Across Devices	36
The Guardian	36
Smashing Magazine	40
Lookout	43
Hulu Plus	45
 Related Concepts: Continuous & Complementary UX	 48
Continuous Experiences	48
Complementary Experiences	50
 Content Remains King	 52

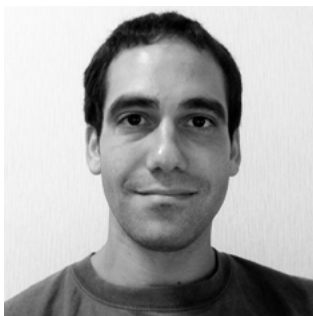


Jerry Cao is a content strategist at UXPin where he gets to put his overly active imagination to paper every day. In a past life, he developed content strategies for clients at Brafton and worked in traditional advertising at DDB San Francisco. In his spare time he enjoys playing electric guitar, watching foreign horror films, and expanding his knowledge of random facts.

[Follow me on Twitter](#)



Ben Gremillion is a Content Designer at UXPin. Previously, he was a Design Writer at ZURB. He started in newspaper design, saw a digital future, and learned HTML/CSS in short order. He later earned an Adobe Certification, and learned up regex, PHP, MySQL, and other impressive-sounding acronyms. He also builds and maintains a CMS for [webcomic artists](#), and participates in bi-annual NaNoWriMo challenges.



With a passion for writing and an interest in everything anything related to design or technology, Matt Ellis found freelance writing best suited his skills and allowed him to be paid for his curiosity. Having worked with various design and tech companies in the past, he feels quite at home at UXPin as the go-to writer, researcher, and editor. When he's not writing, Matt loves to travel, another byproduct of curiosity.

M-Dot Sites Are Dead

New devices come and go every day. The only future-proof strategy is designing fluid experiences that adapt to any device.

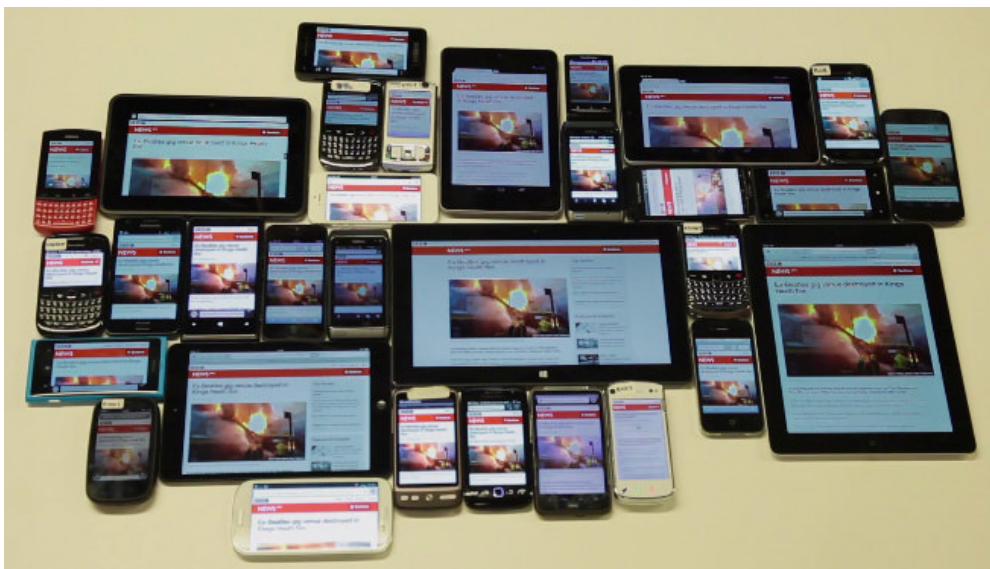


Photo credit: "[Prioritizing Devices: Testing and Responsive Web Design.](#)"
Tom Maslen. Smashing Magazine.

Device consistency is a philosophy that covers principles including creating the correct UX across devices, adopting responsive or adaptive design, and designing around content. We'll explain each of those ideas, but first we'll explain why 2015 signals the decline of M-dot sites.

Back when mobile browsing was new, M-dot sites made a lot of sense. Their faults could have been chalked up to inexperience – we were scrambling to keep up with users and didn't know any better.

Some years later, though, we certainly know better. Many more effective strategies exist, not to mention that mobile devices became more complex, thanks to tablets and varying screen sizes.

This isn't just our speculation, either. [Pure Oxygen Labs reports](#) that last year M-dot sites fell 20%, from 79% in 2013 to 59% in 2014, while responsive and adaptive (dynamic serving) sites rose 37% collectively. At the current rate, M-dot sites are sure to be the minority by the end of 2015, and nothing but a fossil by the end of 2016.

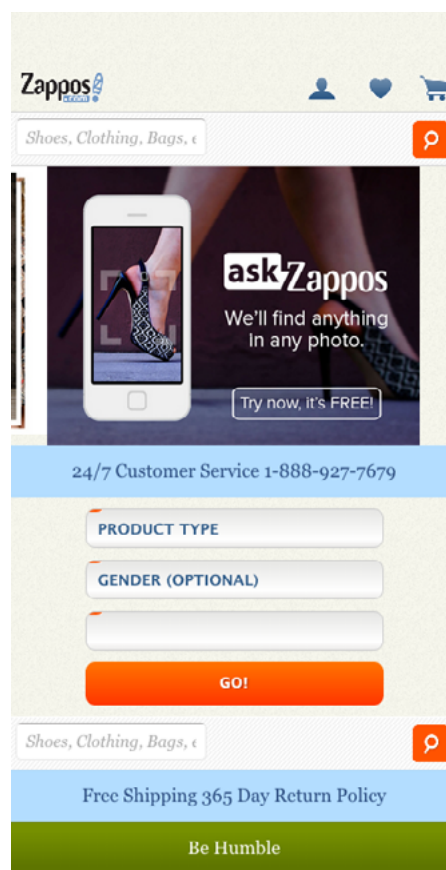


Photo credit: [Zappos](#)

And good riddance, frankly. There's plenty of reasons to abandon M-dot sites:

1. **Users visit the full site anyway** – [Web Performance Today's](#) research showed that about a third (35%) of users choose to go to the full site if given the option.
2. **Users spend more time on the full site** – The same research states 5.5 times longer. They also calculated that 79% of revenue from mobile sales came from users on the full site.
3. **SEO/Google trouble** – According to [Google's own guidelines](#), responsive and adaptive sites will likely rank better. Not using an M-dot is a automatic boost in SEO.
4. **Redirect time** – While M-dot sites load faster in theory, the extra time of redirecting from your full site to the M-dot (unless the user types the M-dot's URL) is unnecessary. Alongside the other drawbacks, is it worth it?
5. **Expensive maintenance** – When you add an extra codebase, you also add more maintenance cost in the long-run. You'll either need to deal with twice the work or use a server-side solution, both of which are more expensive than a responsive or adaptive site.
6. **Mobile devices aren't a single screen size** – It's ironic that what was once the greatest strength of mdot sites is now its greatest

weakness. Mdot sites are designed for a specific screen size, but mobile devices range from 320×240 for some smartphones up to 768×1024 (and beyond) for tablets. It just doesn't make sense to serve the same layout to all those screens.

Bottom line: M-dot sites are a bad idea because they cost more and create inconsistent experiences.

Consistent Experience Across Devices

Providing a consistent experience is the heart and soul of device agnosticism. As we mentioned in the last chapter, the UX is what appeals to people – while UI must adapt to different devices, the UX can, and should, remain rock solid.

Why?

Most people’s perception of “mobile users” is a bit off, perhaps because of the word “mobile.” But, [according to Anna Dahlström’s GeekGirl presentation](#) (which sourced [Think with Google](#)), 77% of mobile browsing occurs at home. This suggests that mobile browsing isn’t out of necessity, but convenience.

It also suggests that most mobile browsing isn’t down in place of desktop browsing, but in conjunction with it. In fact, [the same presentation](#) cited that 90% of users started a task on one device and completed it on another. For example, a person might idly wonder about buying a product, check what’s available on their smartphone

because it's convenient, and then when they get serious switch to desktop to complete the transaction.

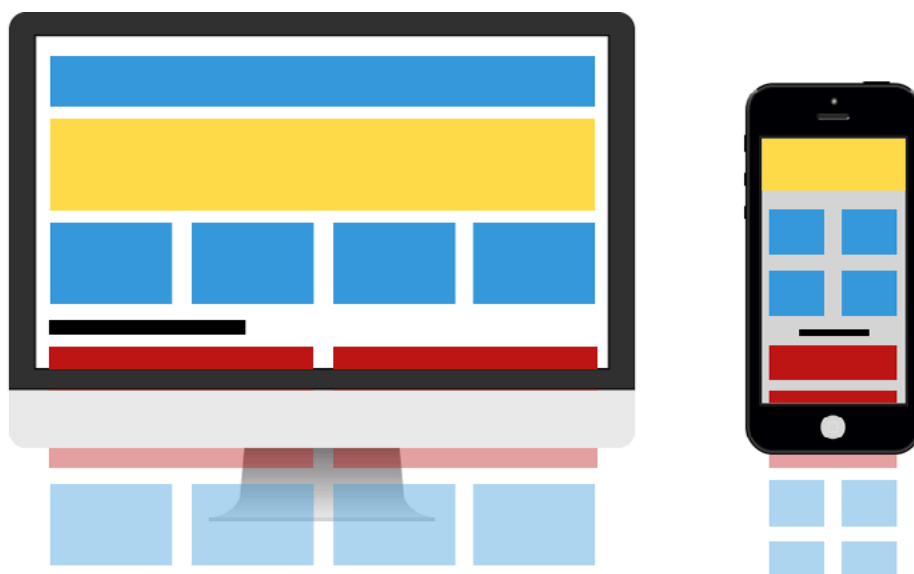
From these statistics, it's easy to understand the importance of a consistent experience, not just as an extra consideration, but for completing a conversion or sale.

How?

Let's examine some strategies to achieve more consistent designs across devices.

1. Visual Consistency.

On the surface level, visual consistency ensures that your site maintains its personality and tone regardless of device. The same colors and graphics, coupled with the same tone of voice, helps create a familiar experience wherever users log in.



2. Flexible Layout.

When working on web designs, ensure that the layout scales appropriately. Obviously, the same site doesn't (and absolutely shouldn't) look the same on every device. But the relative placement of menus, search functions, and key calls-to-action (like logins) should match across devices.

Users become accustomed to location quickly, and don't want to relearn or switch their mental mapping.

For practical advice, try sketching out the ideal layouts for the different devices and comparing them for similarities. [Dahlström's slideshow](#) (mentioned above) gives a thorough explanation of a modular approach to device-agnostic design.

3. Focus on context.

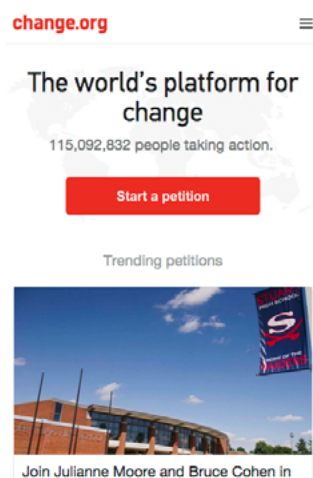
Content suitable on the desktop is not always suitable for mobile. For instance, navigation can be stripped down in a mobile view to simple labels (or even shelved away in a navigation drawer), but they should certainly be fleshed out as you scale up to a tablet or desktop (e.g. horizontal or vertical menus).

When you keep context in mind, you ensure that the design isn't just consistent but also *appropriate*.

Take a look at the progressive view of [Change.org](#) (below).

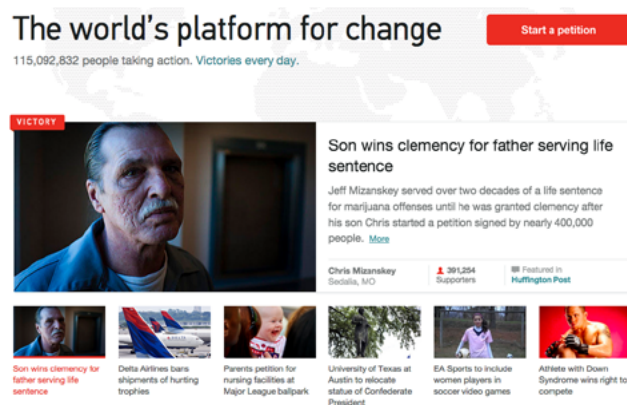
Smartphone View

As a service dedicated to helping people sign and create petitions, it makes sense that the mobile view focuses almost entirely on that goal. The navigation is neatly tucked away and the call-to-action and content stream take priority for people to either browse relevant petitions or create one right away.

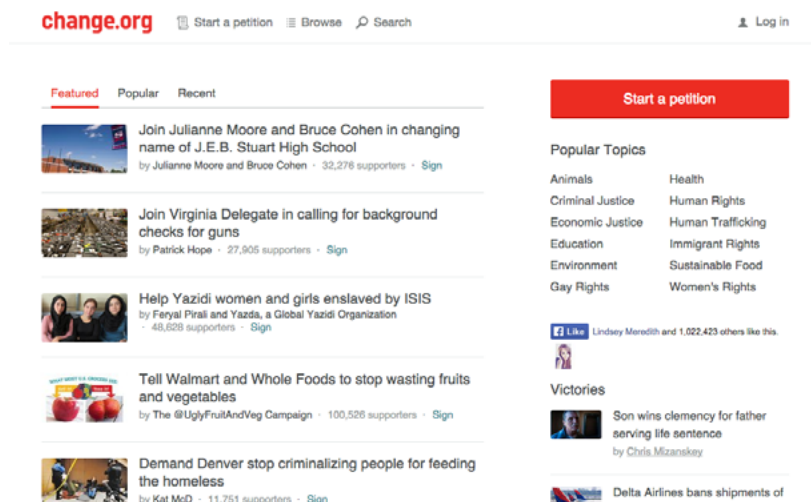


The layout matches how users will likely interact with the site. Either tap to create a petition, or scroll down to find one that matches your interests. Even if you tap the hamburger menu, you can only “Log In” or “Start a Petition”. Simple design for simple user flows.

Tablet View

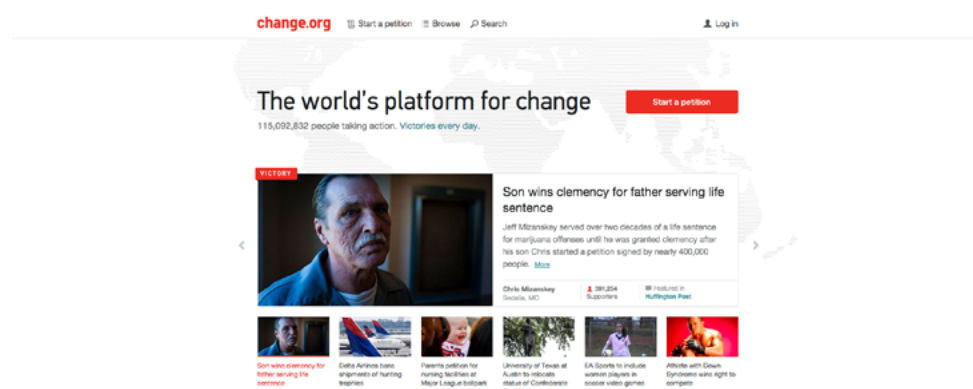


As we move to the tablet view, the navigation expands and the layout shifts to draw attention to featured petitions. Notice how the “Browse” function is now available. The option makes sense for tablet since the mobile screen can’t support such a large list of content.



Desktop View

Finally, you see how the desktop view adds a list of Topics, which makes sense since people would likely click them with a mouse. Add that list to a tablet view (or even worse, a smartphone view) and it would be a nightmare from an interaction design standpoint. Imagine trying to tap each of the tiny links.



Across all 3 views, the core content remains the same and the user goals are always within reach. The differences lie in how supplementary options are incorporated for device contexts.

Responsive and Adaptive Design

When designing sites for device agnosticism, you need to alter to your approach. In recent years, two methods have emerged to address the problem:

- **Responsive Design (RWD)**– Designing a site or app with certain properties (i.e., fluid grid layouts) that allow it to work on all devices.
- **Adaptive Design (AWD, or Dynamic Serving)** – Designing a site with conditions that change depending on the device (i.e., multiple fixed width layouts).

Not only do these two methods safeguard against your product falling flat on certain devices, they also fit perfectly with the mobile-first approach, which we'll discuss below.

Responsive Design

Flexibility is the name of the game for responsive design. Everything must be flexible: layouts, image sizes, text blocks – everything. This

malleability, combined with smart use of CSS media queries, gives your site the fluidity it needs to fit inside any container.



Photo credit: [A List Apart: A Flexible Grid](#)

[A List Apart](#) provides a sample responsive design page, [The Baker Street Inquirer](#). You can play around with the site on different devices to see how it changes.

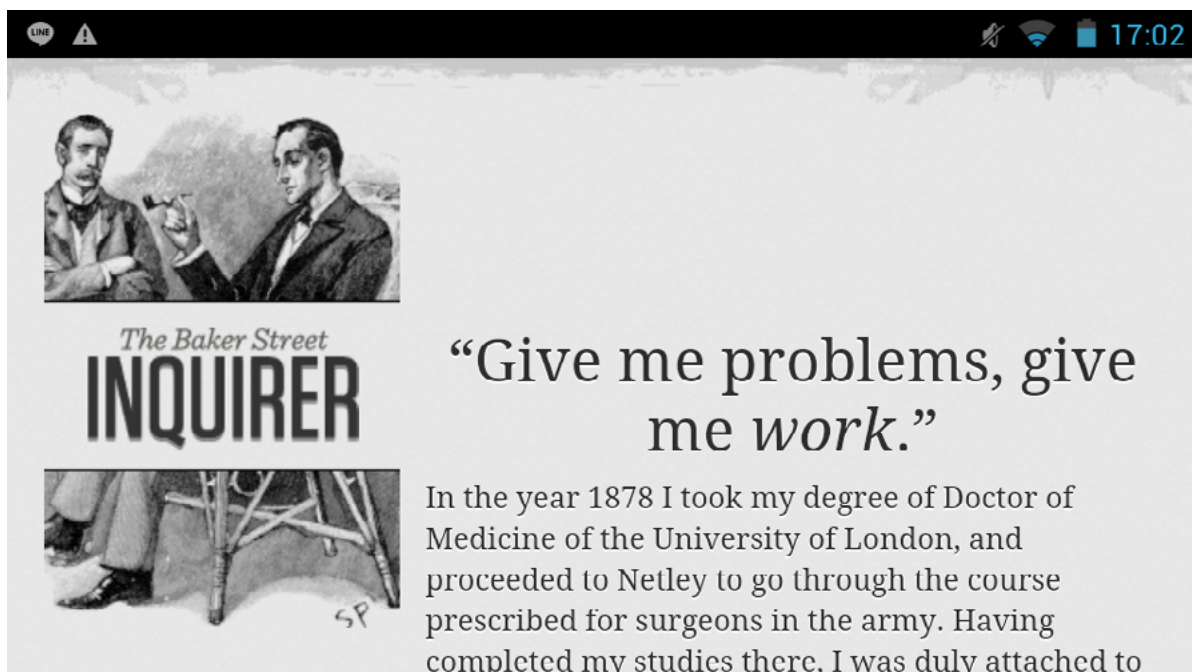


Photo credit: [A List Apart: A Flexible Grid \(mobile\)](#)

RWD gets its name because it responds to the device. Whatever size or system it uses – or even the same device’s landscape or portrait view – the site shifts the layout appropriately. Building a site in this way, too, consolidates your work into just one project to manage.

RWD Best Practices

1. **Flexible images** – Both integral and tricky, fluid images are non-negotiable for responsive design – not only for size, but for cropping (notice how the logo for The Baker Street Inquirer is slightly cropped for the mobile version). For help on the finer points, [Ethan Marcotte gives a thorough tutorial](#).
2. **Scalar Vector Graphics** – When you can, use [SVGs](#) for the best clarity no matter the device. Unlike raster graphics, SVGs alter their resolution based on image paths, not pixels, so they remain the same at any size.
3. **Pay attention to breakpoints** – Knowing breakpoints are the technical requirements for a successful responsive design. For a quick reference guide, read [Media Queries for Common Device Breakpoints](#).
4. **Card Interfaces** – The [card UI pattern](#) can save a lot of headaches since the rectangular shapes act as “content containers” that are easier to shift around.

5. **Keep only what is necessary** – A helpful best practice for any site, but especially in RWD. Think of it like trying to fit the same amount of luggage into different sized suitcases – the less luggage you have, the easier it will be. Responsive-friendliness (and performance) is actually part of the reason why [minimalist interfaces](#) are so popular nowadays.
6. **Prioritize and hide content appropriately** – Desktop screen sizes offer breathing room that smartphone screens do not. Take advantage of hidden controls. For example, to reduce the number of elements that require restructuring, try [progressive disclosure](#).
7. **Large clickable area for buttons** – Fitts’s Law (explained in [Interaction Design Best Practices: Book I](#)) states that the larger clickable area in a button, the easier the user can interact with it. This holds extra weight when the button size fluctuates.
8. **Account for gesture and hover features** – Gesture and hover features are unique to their respective devices, but users love them because they enhance the experience. Factor both into your design, for example, a hover animation on desktop can become a touch animation for mobile.

For more information, including advice and help with the coding, read [Kayla Knight’s guidelines for responsive design](#). Or, if you want to jump right in but need help with CSS, check out these CSS media query guides below:

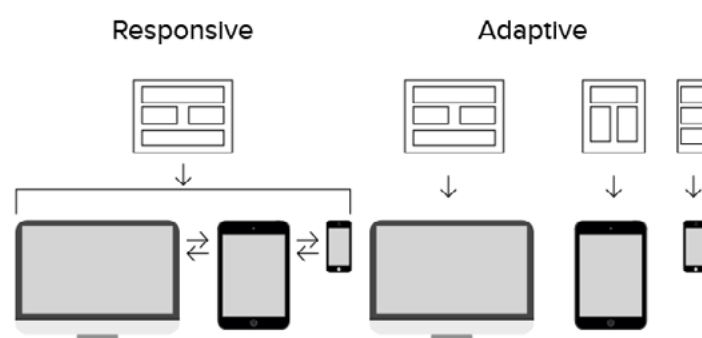
- [Use CSS media queries for responsiveness](#) – Pete LePage
- [Media Query-less Design, Content-based Breakpoints & Tweak-points](#) – Dave Olsen

Adaptive Design

For more control than RWD sites, some designers prefer adaptive design, where they essentially design different sites for different categories of devices. Typically AWD sites have up to six variations, based on screen width:

- 320
- 480
- 760
- 960
- 1200
- 1600

With AWD, functionality plays a bigger role. Designers can create entirely new interfaces around a device's attributes, such as emphasizing touch controls, or the large canvas for desktop backgrounds.



AWD sites also reduce loading time.

Think about it: if you design a mobile site without any of the elements a desktop site requires, there's just less to load. In fact, [Catchpoint tested loading times](#) in WordPress for a responsive theme and an adaptive one (using [Wiziapp](#)). The results favor AWD:

Metric (Defaults)	Adaptive	Responsive
Response	568 ms	1,202 ms
Document Complete	1,536 ms	4.086 ms
Webpage Response	2,889 ms	4,860 ms
Bytes Downloaded	2,474,326 kb	4,229,362 kb
Objects Downloaded	20	61

Photo credit: [UXPin Blog](#) via [Catchpoint](#)

AWD Best Practices

- **Maintain consistency** – Don't let the freedom go to your head. Remember that a consistent UX across devices is crucial, so maintain a common thread between devices, such as the general relationship between elements.
- **Use a grid** – A 12-column framework is preferred with consistent margin and gutter widths that align to a baseline grid. Grids can be full-width or centered.

Responsive web design is better for the user since the experience is tailored specifically for their view, but adaptive web design isn't

a bad compromise at all if you're on a time crunch and still want a mobile-friendly experience. For example, designing for 3 device breakpoints is better than a desktop/mdot site approach and requires less work than responsive. It won't create as fluid of an experience, however.

Mobile-First Design

The mobile-first approach is exactly as it sounds: designing for the smallest screen and working your way up. It is one of the best strategies to create either a responsive or adaptive design.

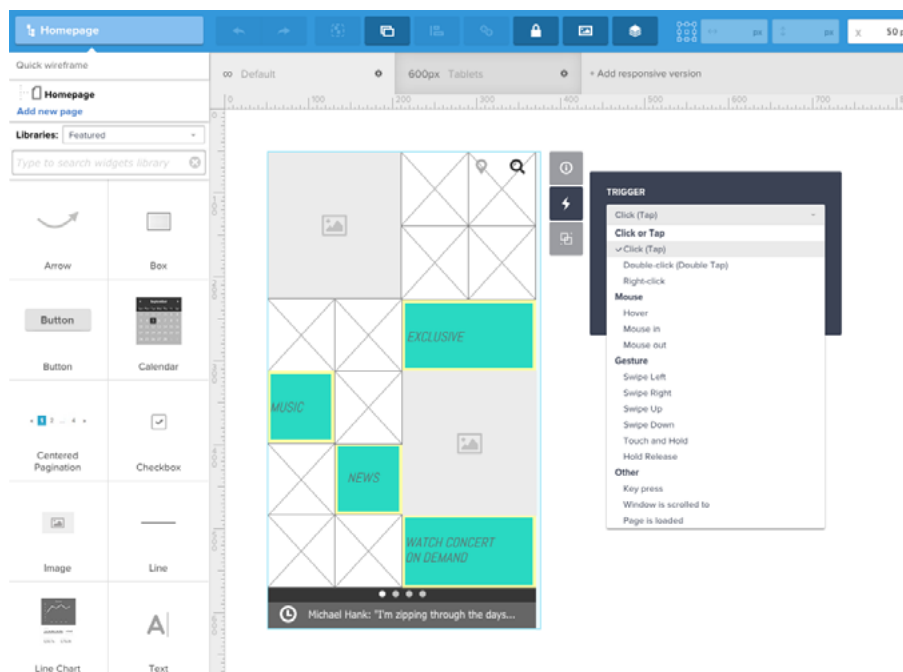


Photo credit: [UXPin](#)

- The mobile-first approach is a tenet of **progressive enhancement**. It is the ideology that mobile design, as the hardest, should be done first. Once the mobile design questions are answered,

designing for other devices will be easier. What it boils down to is that, the smallest of the designs will have only the essential features, so right away you have designed the heart of your UX.

- **The opposite approach is graceful degradation.** This incorporates all of the complexities right from the start, then strips them away later for smaller devices. The problem with graceful degradation is that when you build the all-inclusive design right from the start, the core and supplementary elements merge and become harder to distinguish and separate. The entire philosophy runs the risk of treating mobile design as more of an afterthought since you're "cutting down" the experience.

We, [along with many others](#), strongly recommend progressive enhancement with a mobile-first approach.

Mobile-First = Content-First

If your site is good on a mobile device, it translates better to all devices. More important, though, is that the mobile-first approach is also a content-first approach. Mobile has the most limitations, screen size and bandwidth to name a few, and so designing within these parameters force you to prioritize content ruthlessly.

The mobile-first approach organically leads to a design that's more content-focused, and therefore user-focused. The heart of the site is content – that's what the users are there for.

One caveat, though, is that mobile users sometimes require different content than desktop users. Device-specific content can be gauged by considering context – what, in a given situation and a given environment, will your user appreciate more. The best way to plan ahead for these is creating user scenarios.



Photo credit: [UXPin](#)

Another advantage to mobile-first approach is that the small-screen breakpoints can better fit around the content. Again, the alternative is worse: having to squeeze an already plump design into a tiny framework. But with the mobile-first approach, the breakpoints develop naturally around content, so you don't need any awkward edits.

The Mobile-First Process

We'll describe a process that helps our designers at [UXPin](#).

As usual, wireframing is a recommended early step to most efficiently structure your layout. When wireframing or prototyping, we use the [responsive breakpoint menu](#) streamlines the process of moving to different screen sizes, starting with the smallest.

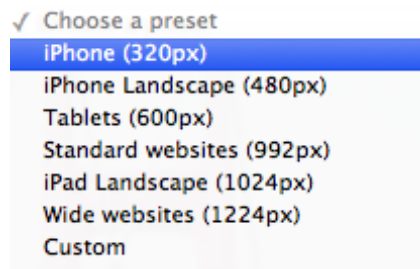


Photo credit: [UXPin](#)

These presets layout the proper screen size for you, so you can wireframe keeping only the content in mind.

Our procedure follows these steps:

- 1. Content Inventory** – This is a spreadsheet or equivalent document containing all the elements you want to include.

	Navigation title	Page title	Files	Last updated	Owner	Comments	Delete?
0.0	Home	Wine Tasmania					
1.0	Wine Tasmania					No page at this level - displays 'History'	
1.1	History	History					
1.2	Touring Tasmania	Touring Tasmania					
1.3	Touring Links	Touring Links					
1.4	Wine Industry Tasmania	Wine Industry Tasmania					
1.5	Industry Statistics & Info	Industry Statistics & Info					
1.6	Investment	Investment					
1.7	Partners	Wine Industry Tasmania Partners					
2.0	The Wine Route					No page at this level - displays 'Overview'	
2.1	Wine route overview	The wine route					
2.2.0	Tamar Valley Wine Route	Tamar Valley Wine Route					
2.3.0	Southern Wine Region	Southern Wine Region					
2.4.0	East Coast Wine Region	East Coast Wine Region					
2.5.0	North West Wine Region	North West Wine Region					
3.0	Latest News	Latest News				No content on page	
4.0	Events					No page at this level - displays 'Overview'	
4.1	Overview	Events				No left-nav	
4.2	Booking	Event booking				No left-nav	
4.3	Privacy Policy	Privacy Policy				No left-nav	
4.4	Security and Refunds	Security and Refunds				No left-nav	
5.0	Members	Wine Industry Tasmania Members					
6.0	Resources	Resources	5 PDF files				
7.0	Contact Us	Contact Us				Email address & contact form	

Photo credit: [Maadmob](#)

- 2. Visual Hierarchy** – Prioritize the elements in the content inventory and determine how to display the most important elements prominently.

3. **Design with the smallest breakpoints and then scale up** – Build the mobile wireframe first, then use that as the model for larger breakpoints. Expand the screen until there's too much white space.
4. **Enlarge touch targets** – Fingers are much wider than pixel-precise mouse cursors, and so need larger elements on which to tap. At the time of this writing, Apple recommends 44 x 44 points square for touch targets. Give hyperlinks plenty of space and slightly enlarge buttons to ensure that users don't need to tap twice.
5. **Don't count on hovers** – It almost goes without saying, but designers often rely on hover and mouseover effects in their interactive work. If you're thinking mobile-friendly, don't. There is no hover control for fingertips yet.
6. **Think “app”** – Mobile users are accustomed to motion and a modicum of control in their experience. Think about [off-canvas navigation](#), expandible widgets, AJAX calls, or other elements on the screen with which users can interact without refreshing the page.
7. **Avoid large graphics** – Landscape photos and complex graphics don't display well when your screen is only a few inches across. Cater to mobile users with images that are readable on handheld screens.

8. **Test it in a real device** – Nothing beats discovering for yourself how usable a website is (or isn't). Step away from your desktop/laptop computer and load up your product on a real phone or tablet. Tap through pages. Is the site easy to navigate? Does it load in a timely fashion? Are the text and graphics easy to read?

This is just a basic outline. For the complete guide to our process, download the free [*Content Wireframing for Responsive Design*](#).

A Mobile-First Design Lesson

Given that different devices need different layouts based on their screen size and orientation, it makes sense to design multiple arrangements for your users. Luckily you can make your own responsive or adaptive variations right in [UXPin](#).

We'll create an example and describe how to scale up content from a smartphone to the tablet and desktop views.

Set your content priorities

A “mobile-first approach” differs from “desktop-first” in that we add information to each progressively-larger layout rather than cut away as we design smaller. Thinking mobile doesn't mean eliminating information. It means sorting information into primary, secondary and tertiary content.

In this example, we know that the home page should have certain elements, like the company's name and links to products. A blog post

wouldn't hurt either. But like we said, not everything will fit into a smartphone view, so we set priorities based on what will achieve the site's goal: selling bikes:

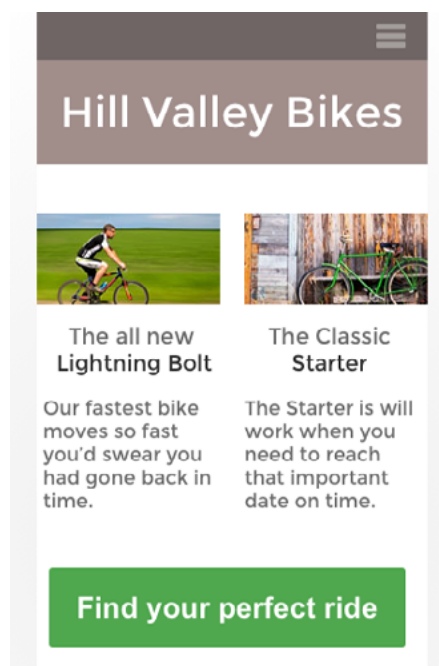
1. The newest model bike
2. The best-selling bike
3. "Find your perfect ride" CTA
4. Company name and hero image
5. Navigation
6. Search
7. The second-best-selling bike
8. Gift certificates
9. A testimonial
10. The latest blog post

Based on that ordered list, we can create with the confidence that our work will solve a design problem of getting sales.

Smartphone View

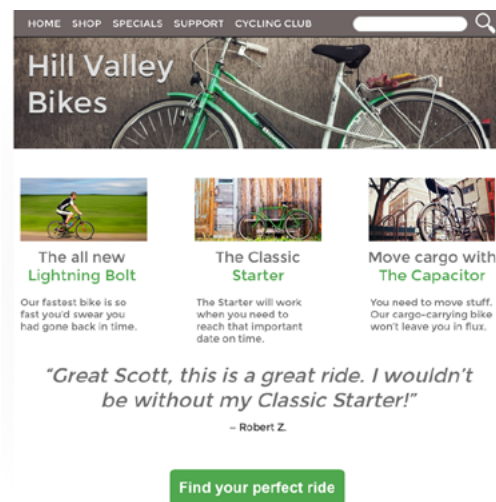
How much do users need?

Thinking mobile-first forces us to think about what's *really* important. In this smartphone view, the top-selling bike and newest model will lead directly to sales, so can we leave other items – such as gift certificates, a less-popular model, the latest news – for inside pages. The final call to action is especially prominent and easy to hit with a single tap of the finger.



Tablet View

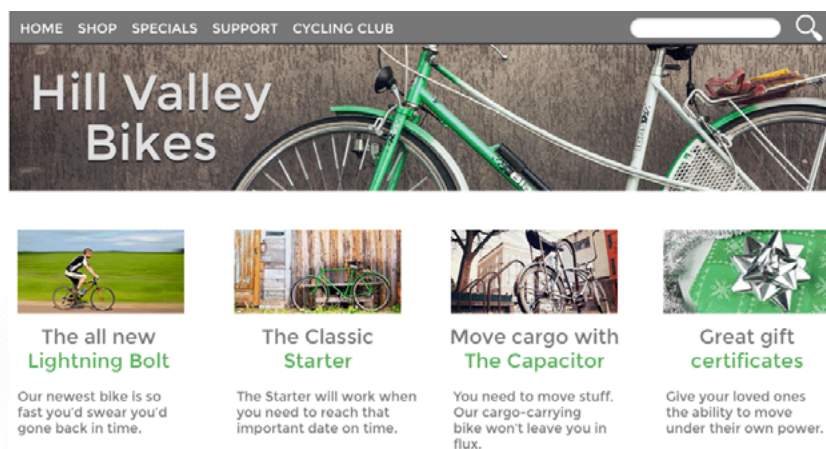
As we design for a tablet-sized view, we're better able to add secondary information like additional products (e.g. "The Capacitor"). We can also expand the navigation at the top of the page and add content that encourages sales without actually leading to them – namely, the testimonial.



Because more options are available, this can be surprisingly more difficult than deciding what to include in a smartphone UI. The difference between secondary and tertiary elements is a blurry line, and temptation is strong to include everything.

Resist the urge. Use the ordered content list. Like smartphones, space is still limited.

Desktop View



Finally, the desktop view can support as much information as you decide is important. This is where the home page can accommodate

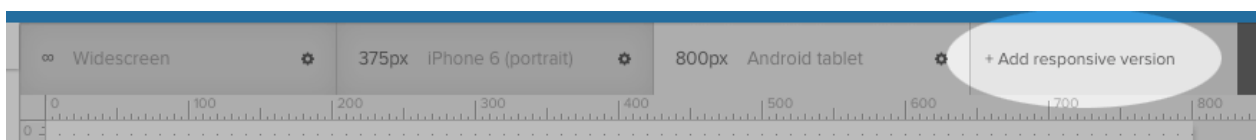
all of the information you see fit, whether or not it fits. Notice some of the additional content we've included:

- Gift certificates
- Customer testimonials
- Blog post exploring the newest Lightning Bolt bike

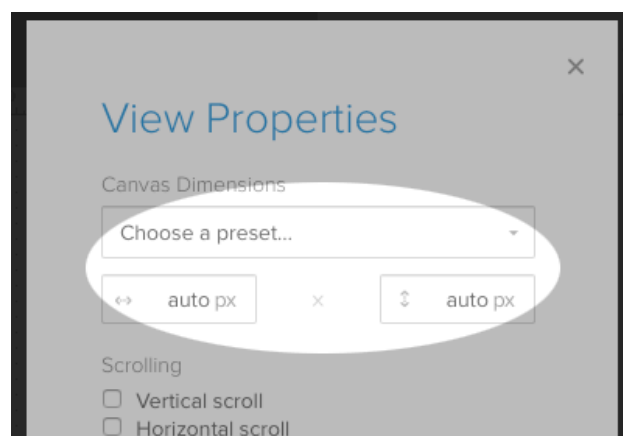
Design device-appropriate layouts yourself

If you're using [UXPin](#), it's fairly easy to create different layouts for these views.

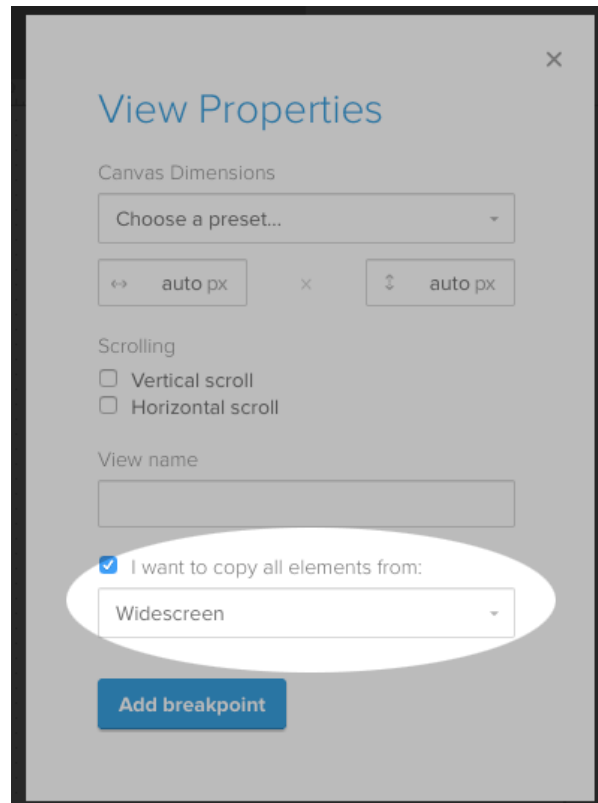
1. Open a UXPin prototype.
2. Tap “add responsive version” at the top of your prototype's canvas.



3. Choose a preset size or enter your own dimensions.



4. You don't have to recreate everything from scratch. Choose a size from which to copy your design's elements.



And that's it. Switch between breakpoints by tapping the different sizes above your canvas, and adjust each to suit your needs.

Real Examples of Consistent UX Across Devices

The Guardian

The site for the famous British newspaper [The Guardian](#) is a great example of mobile-first device consistency.

In keeping with our own advice, let's start the analysis with the smallest screen:

Smartphone View



Photo credit: [The Guardian](#)

The smartphone view is cohesive and inviting, with all the essential elements presented in a clear visual hierarchy.

- Right at the top, the necessities are in the banner, with login, search, and the site's title.
- Directly below are the most popular navigation categories (home, "US," "world," etc.) for easy access. Additional categories are hidden in the hamburger menu (following the principle of progressive disclosure). It's a fine balance because placing all categories into the hamburger menu limits searchability for the whole interface.
- The features story, with its enticing image, takes up most of the room, showing that it's the most important element. And yet, with a quick scroll, the user can access any number of secondary stories. This facilitates browsing, but leaves some control in the hands of the designer.

No space is wasted on the mobile version, too – even the white space opposite the "headlines" title features weather information, a little extra something that's a nice touch.

Let's see how this compares to the tablet version below.

Tablet View

- At the top, the banner remains the same, but the tablet offers more room for additional elements ("jobs" and the country edition), text titles for the icons, and a subheading to the guardian's logo/brand name to give it extra social proof. The hamburger menu

remains, although there are more categories listed than only the most popular.

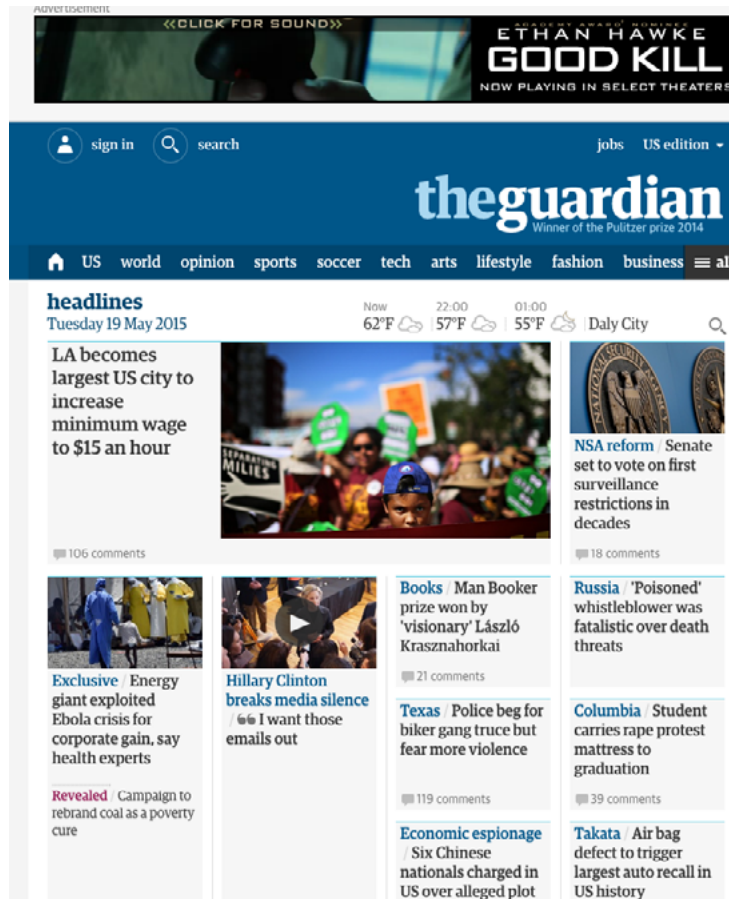


Photo credit: [The Guardian](#)

- The biggest difference is that the tablet offers a lot more stories to choose from, and breaks the single column organization. This creative use of the card UI pattern allows the designers to assign more priority to certain stories (“L.A. becomes...”) using size, while still keeping the site tidy.
- The tablet version can even afford an ad at the top. Moreover, the weather data is elaborated, and there’s even room for the full date, a newspaper staple.

All in all, the tablet version feels like a roomier, more luxurious version of the smartphone screen. How does that look next to the desktop version?

Desktop View

Let's take a look:

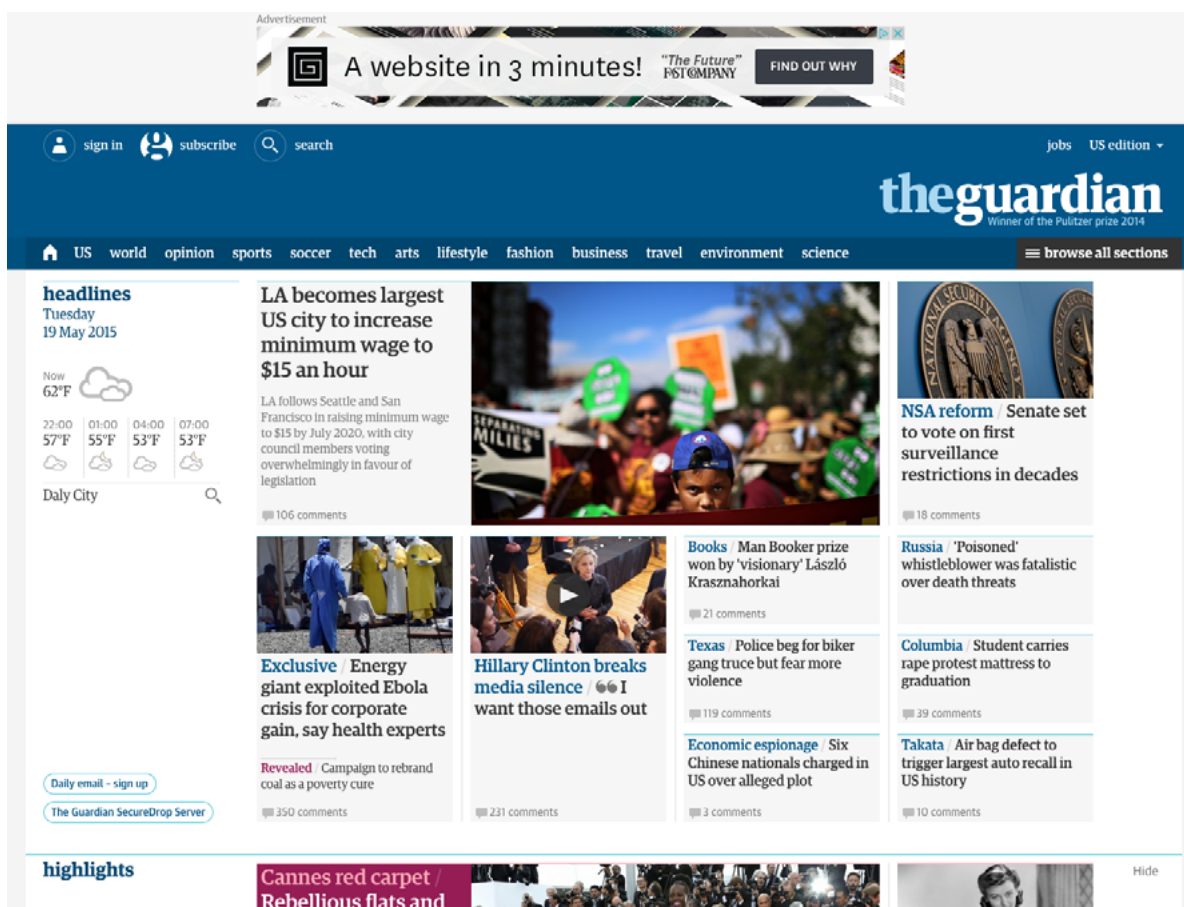


Photo credit: [The Guardian](#)

The full desktop view reveals the true mastery of the site. What the Guardian does right across all devices is consistency – all three sites deliver the same overall experience. All the versions are scroll-based, all use the same style of cards, all have the same brand banner at the top and the key navigation elements in the same places.

The biggest difference is that the larger screens feature more information, from more available story cards to more complete weather details. The core content, however, is accessible on any device.

The site doesn't look the same across each device, but it definitely feels the same. Users familiar with only one version will still feel comfortable using another.

Smashing Magazine

With articles advising designers on how to create better mobile experiences, naturally [Smashing Magazine](#) follows their own advice.

Smartphone View



Smashing Magazine's content is still readable despite the smaller screens, remaining useful to their readers on the go. They know full well that the users will scroll, so they don't "rush" with the content. They space out the full title over four lines, plus byline, date, category, and a link to the comments. Notice how they clearly labeled the navigation menu, instead of just making it a hamburger icon.

Smashing Magazine smartly puts faith in their users and designs for usability, which in this case adds to their UX.

Tablet View

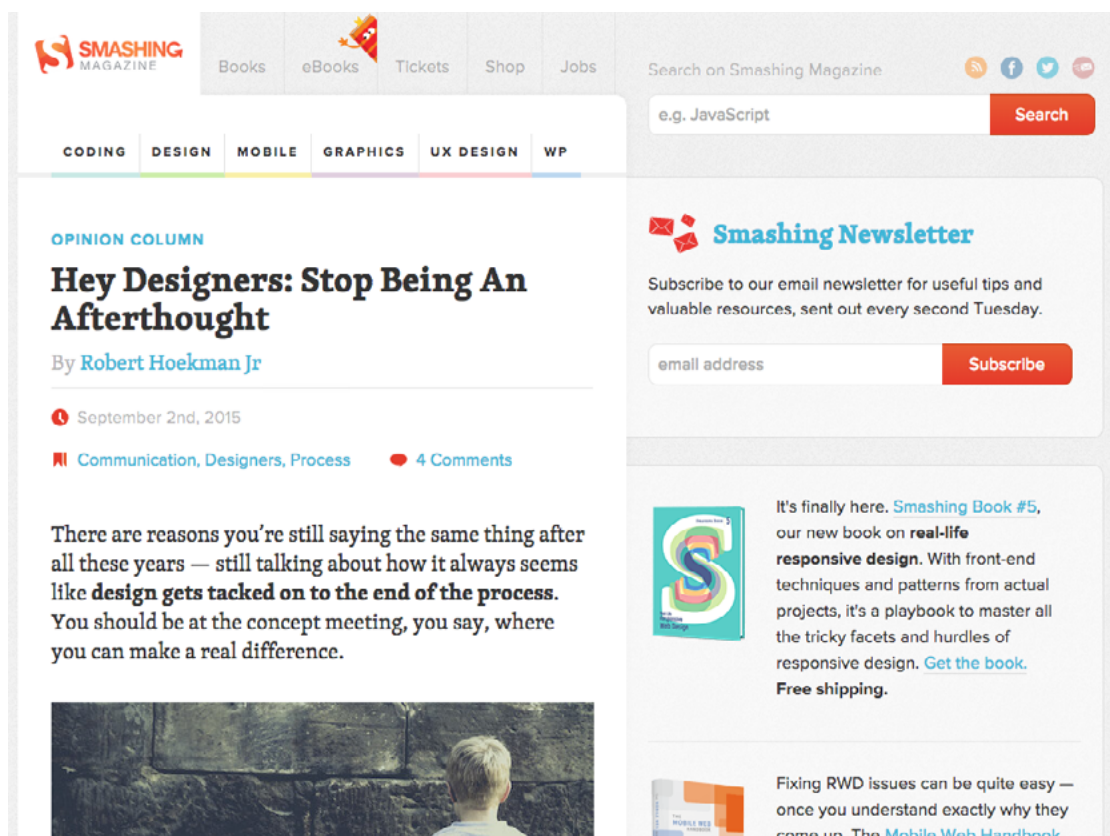


Photo credit: [Smashing Magazine](#)

Content remains the focus of the page, but the wider view allows for the "Menu" button to expand into a full navigation.

Desktop View

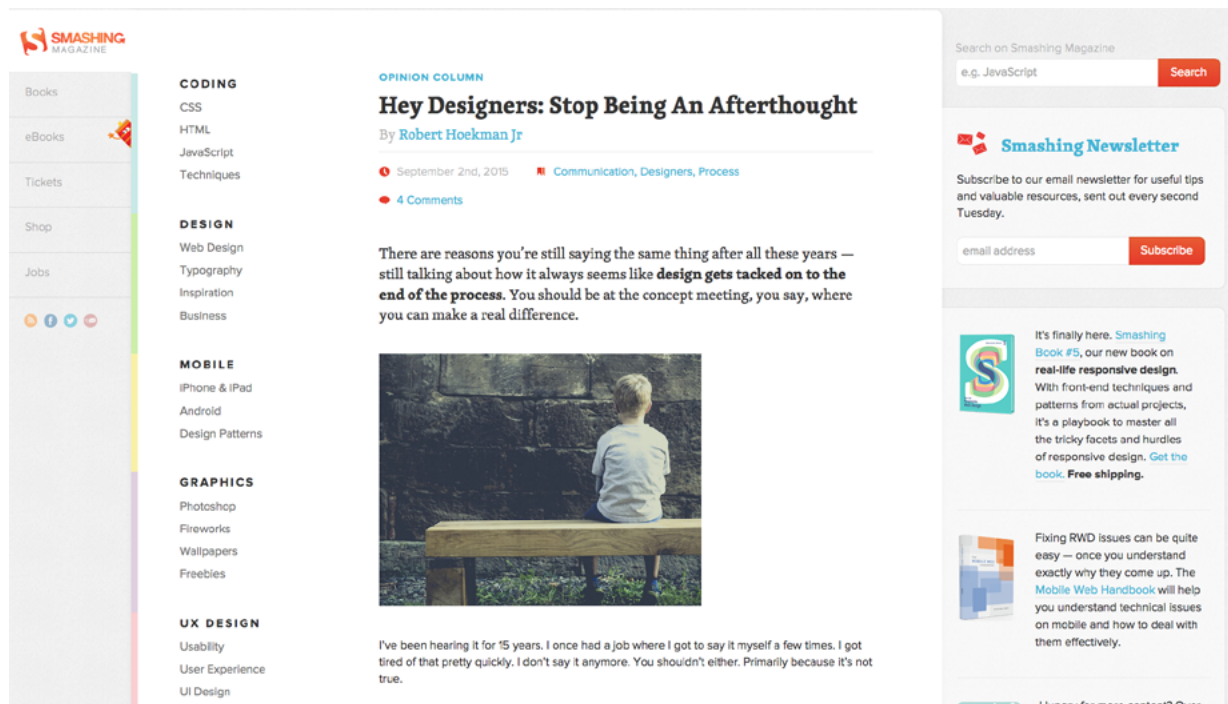


Photo credit: [Smashing Magazine](#)

Last, the desktop view has room to put its feet up. Even using the right quarter for ads and burning the left eighth for white space under its navigation, the site still has room to legibly display the article and images therein.

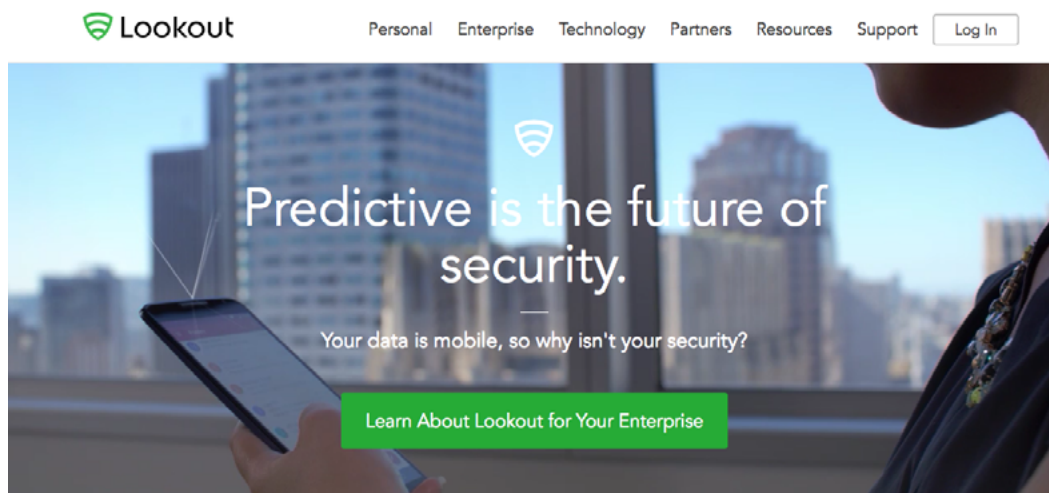
The promo of their categories – crowning the tablet view – now has its own column to the left.

Notice how the crucial information, such as byline and article information, remain more-or-less the same across all three devices. This creates a familiar orientation for regular readers on all devices.

Lookout

The mobile security site Lookout knows the limitations of mobile, and doesn't try to force their desktop UI on it.

Desktop View



For example, their home page features a beautiful hero animation background, that unfortunately would not translate into mobile. Instead, they used a meaningful screenshot – cropped – for their mobile background.

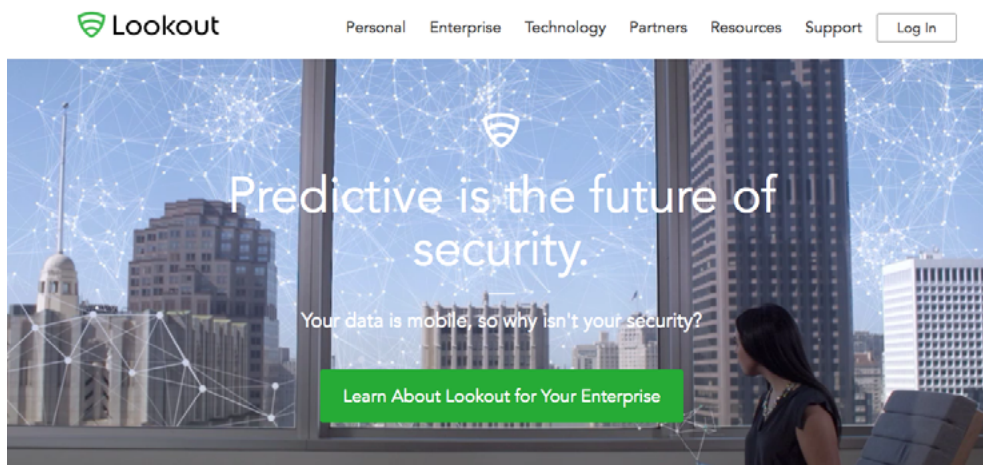
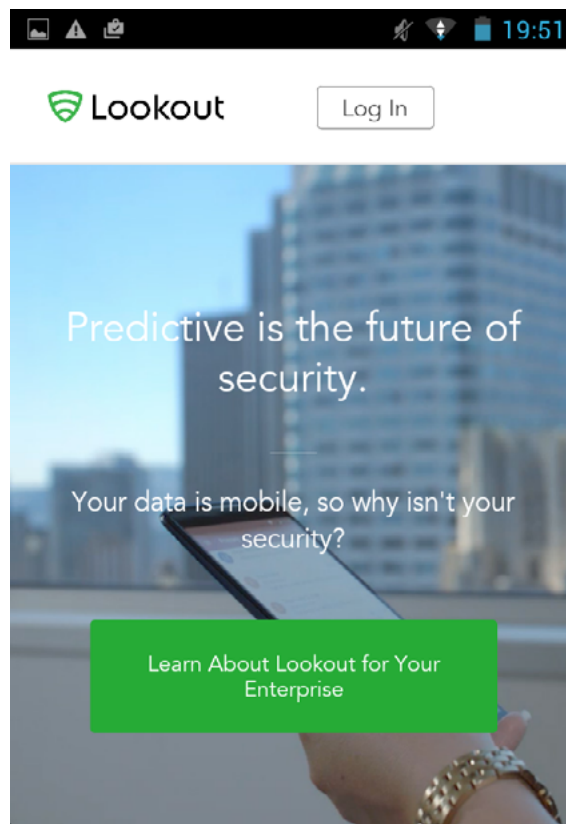


Photo credit: [Lookout](#)

Mobile View



Predictive Security Is What Makes Lookout

Photo credit: [Lookout \(mobile\)](#)

Notice that the other elements remain similar – same text, same layout, same green call-to-action. The mobile header, however, has been reduced to the two essentials: logo and login.

Hulu Plus

Last, [Hulu Plus](#) shows us that maintaining consistency isn't about mirroring the site on all devices.

Native Mobile App

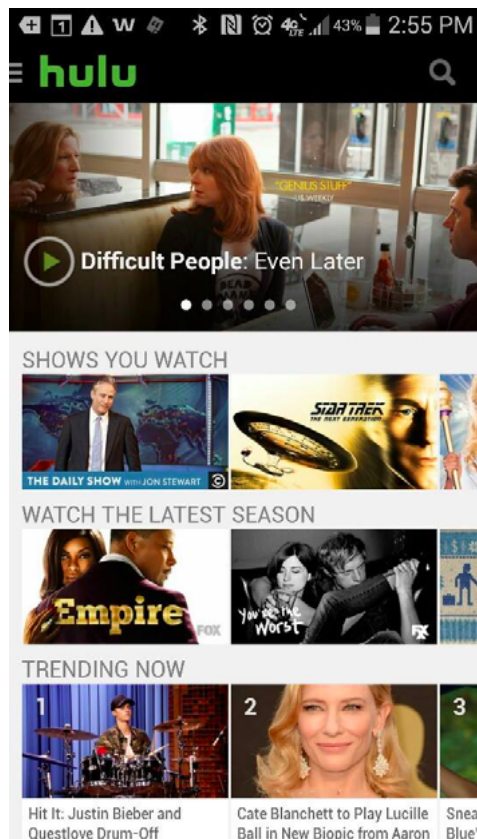


Photo credit: [Hulu](#)

It makes sense for a video streaming service like Hulu to deliver its mobile experience primarily through an app rather than in the browser:

- **Faster load times** – Instead of loading a page from scratch, mobile apps like Hulu's come with pre-loaded components. Every bit of speed helps when you're streaming video on mobile devices

- **Slicker UX** – You don't need to deal with a browser interface, which reduces clutter and opens up more creative options. The experience also feels more self-contained since you're not clicking through pages, waiting for the load screen to fill up, and dealing with permissions requests.

Of course, a mobile app is not always better than a responsive site. For example, small businesses and restaurants don't need mobile apps because they wouldn't deliver any additional value. On the other hand, businesses definitely benefit from native apps if the service requires computational power, advanced account management, and/or relies heavily on streaming media.

Tablet

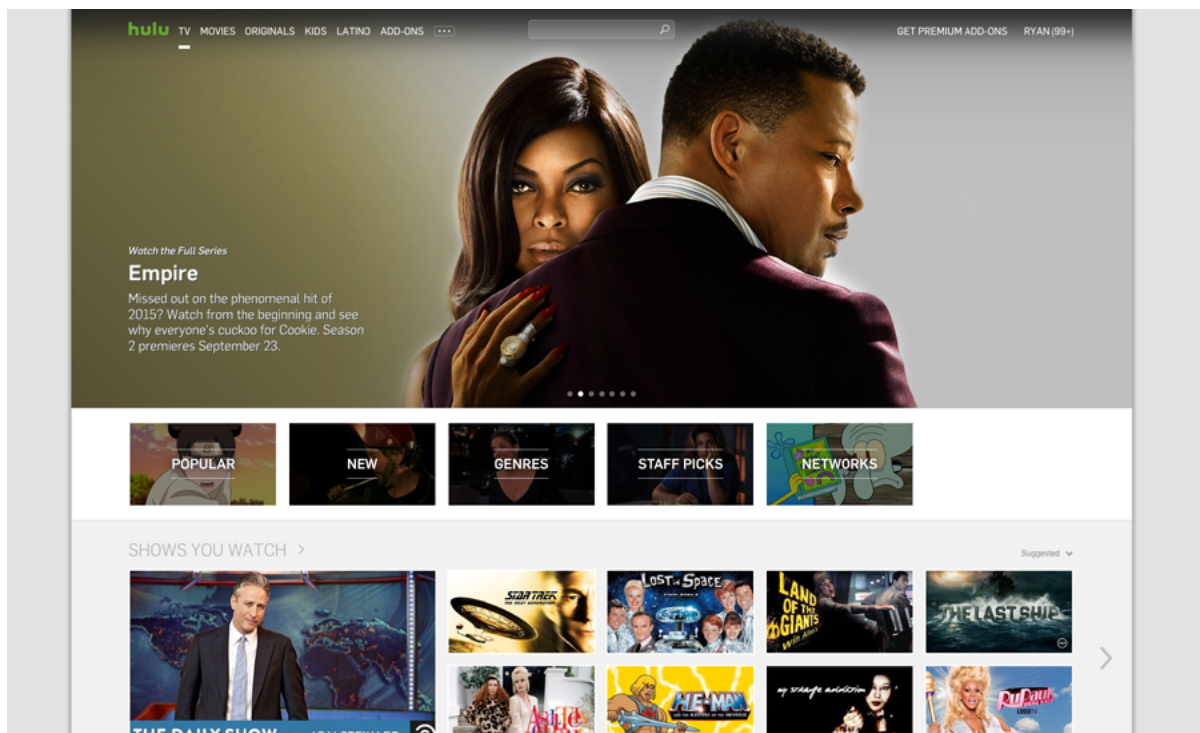
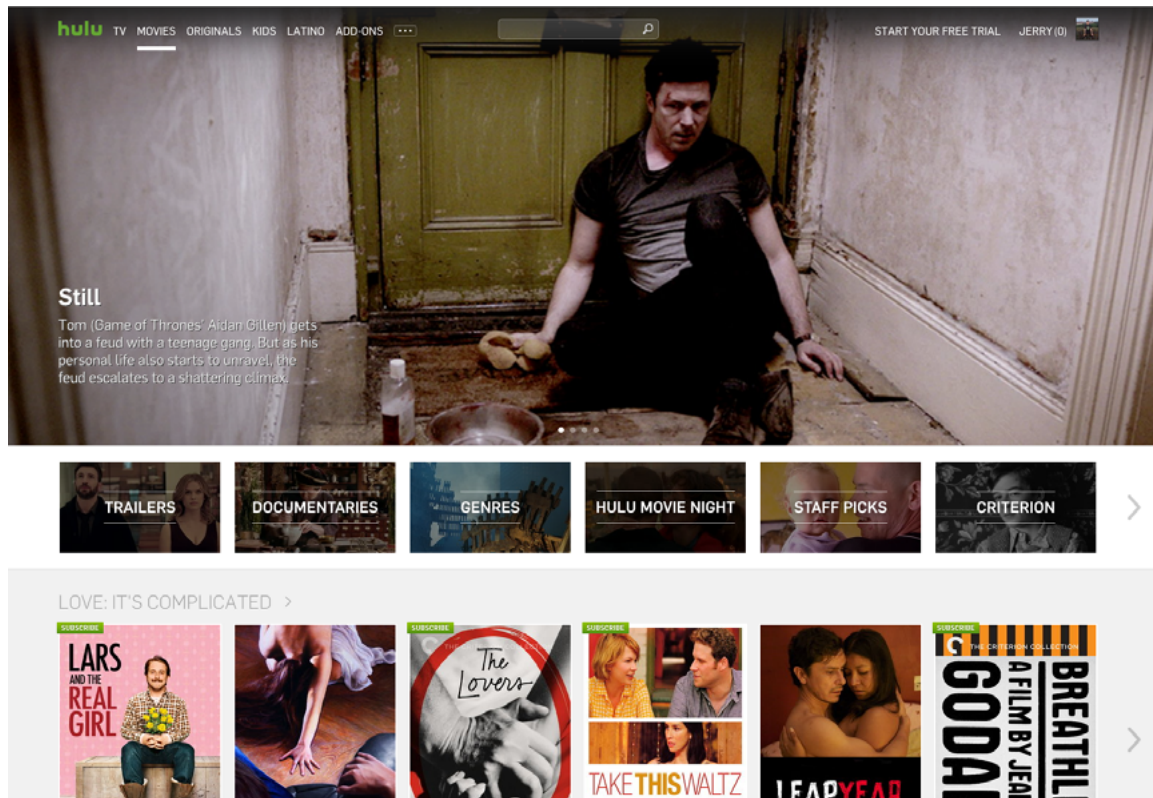


Photo credit: [Hulu](#)

As we expand to a tablet view, additional shows appear near the bottom of the screen. The layout remains largely unchanged. You can see how the cards-based UI scales well to different sizes.

Desktop View



Notice how the layout doesn't change so much as expand. The important features remain with featured content in clear view with suggestions at the bottom.

The site doesn't conform to the same format on different devices, but remains consistent enough that it's still familiar and understandable.

Related Concepts: Continuous & Complementary UX

Now that we've provided an overview of consistent design, let's zoom out for a moment so you can see the whole multi-device landscape.

While we're discussing consistent design here, we've also seen the rise of two related strategies thanks to the work of Michal Levin in her excellent book [*Designing Multi-Device Experiences*](#) (a highly recommended read).

Continuous Experiences

Since users often conduct the same task on different devices, it's important to make the transitions as seamless as possible. For example, you might check your Gmail on a mobile device, start writing a response, get distracted, and later resume the draft on desktop.

You can understand the frustration of a user if they were 90% done with an email written on their mobile device only to have it disappear on the desktop.

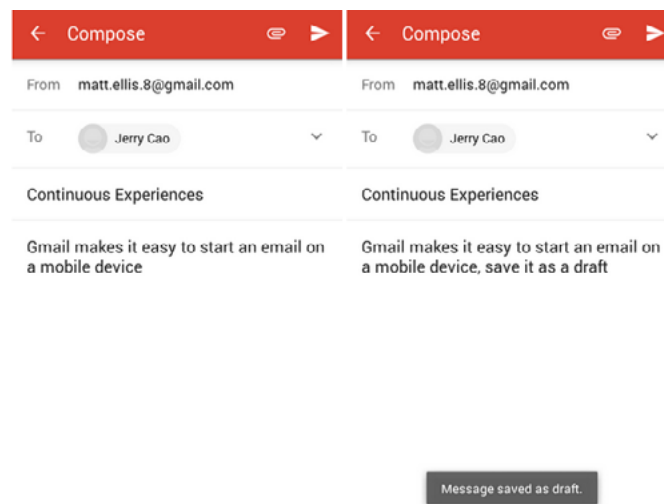


Photo credit: [Gmail \(App\)](#)

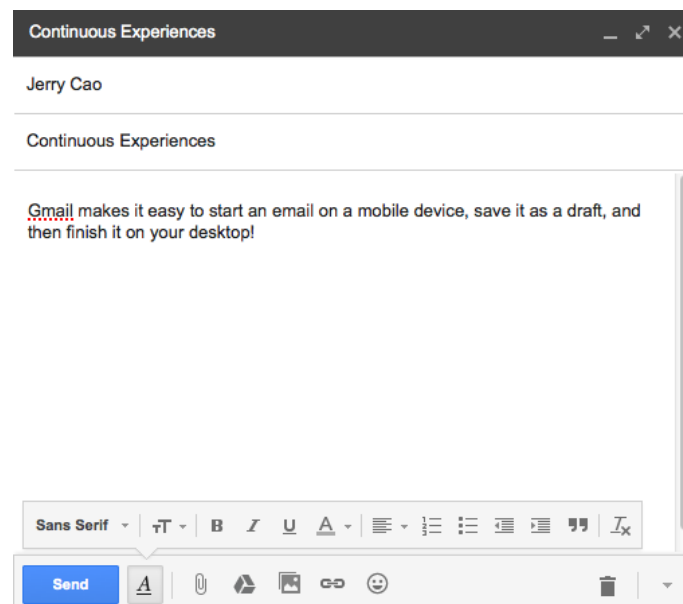


Photo credit: [Gmail](#)

This doesn't just mean continuous between devices, either – it can also refer to going from the digital world to the real one. For example, the [Home Depot site](#) tells you if a product is available in the store nearest you.

Any way you can make your user's experience easier on them, they will repay you with loyalty.

Complementary Experiences

While currently most multi-device tasks are sequential, i.e., one after another, we're starting to see a rise in devices that work simultaneously.

For example, [Cybeer Bar](#) below is a game site in which users control the mug with the motion controls on their smartphone, with the aim of learning how to pour the perfect cup of beer.



Photo credit: [Cybeer Bar](#)

According to UX designer [Michal Levin](#), these complementary relationships in general fall into one of two categories:

- **Control** – One device controls the other, such as using a mobile device to control Netflix on a TV.
- **Collaboration** – Both devices work together with specific roles, such as controls for [Cybeer Bar](#).

Aside from novelty games, companies like [Netflix](#) are also creating complementary experiences by allowing the mobile app to communicate directly with Chromecast. Users are then able to browse content on their phone and watch it on the the TV in front of them.

While this type of device relationship is still fairly new, expect it to become a more popular trend over the next year.

Content Remains King

The main point we hope to convey is that content is king. A phrase [coined by Bill Gates](#), it still rings just as true today as it did almost two decades ago when he said it. Your users are coming to the site for the content – the UX – so this should lead the UI.

Device consistency is a content-first approach: it recognizes that the technical aspects of the device come second to the content that is being displayed. The relationship between device agnosticism and content is a two-way street, though – users value versatility so that they have the option of using your site on whichever device they choose.

For that reason, device consistency and content-first are one-in-the-same: both put the user's best interests first.

Design responsive & adaptive prototypes in UXPin (free trial)

Everything you ever wanted in a **UX Design Platform**

- ✓ Complete prototyping framework for web and mobile
- ✓ Collaboration and feedback for any team size
- ✓ Lo-fi to hi-fi design in a single tool
- ✓ Integration with Photoshop and Sketch

Start using it now!